

SW Engineering CSC648/848 Section 01 Fall 2017
Bay Real Estate, Team 12, Milestone 1
September 30th, 2017

Jason Cromer (jcromer@mail.sfsu.edu)

Artsem Holdvekht

Chen-Feng Huang

David Hoang

Jiawen Zhu

Jordan Leong

| Revision | Revision Notes |
|----------|--|
| 1 | Update Use-Cases and inconsistencies between them and Data Definitions |

Executive Summary

Our website is looking to profit from the unique and always growing real estate market in the San Francisco Bay area. We are going to provide an all in one service for anyone who is interested in selling or buying the real-estate. The service will provide a streamlined and easy to use interface in order to attract potential customers. The system will be able to support unregistered/registered users, real estate agents, and system administrators.

Unregistered **buyers/sellers** will be able to access the full website and view all the property listings that are available at the moment. They will also be able to utilize basic features, such as search and filtering in order to better their experience. The registration process is completely voluntarily and will not be enforced, until the user decides to buy/sell property and view more in-depth information. The interface will be pleasing, easy to use, and require little to no technical experience.

Registered **buyers** will have access to unique features such as ability to save their search and add listing to **favorites**. In order to be able to use that functionality users will have to register and provide their full name, and phone number.

A real estate agent, or **seller**, will have to register to be able to post and edit property listings. The agent will have to provide basic information, such as their real-estate **license number**, in order to verify legitimacy of that person or business. If the user represents a private party as a sale by owner, that user will have to provide a **Driver License number** in order to verify their identity. Upon clearing, the **seller** will have access to the personal information that they agreed to provide when accepting the terms of user policy when signing up for the services. These include items such as **name**, **phone number**, and real estate **license number** or **Driver License number**. Once registering process is complete, these users will be able to post and edit property **listings**.

All the processes will be monitored by company's **admins**, which will be provided with the minimal interface and basic training, if necessary, to edit and monitor the content that is being uploaded to the website and resolve any technical issues that might arise.

The interface for **buyers** and **sellers** will be pleasing and easy to navigate. The ease of use, and appeal will be important in drawing in new customers to the business. We assume the user might have very little to no technical experience, and that will be one of our advantages compared to the competition.

Use Cases

- 1.) **Unregistered Buyer:** Tim is an unregistered **buyer** interested in purchasing a house. Upon opening our website, he can immediately search and browse our page to look for any houses that may interest him. While browsing, Tim stumbles upon an interesting house and can view the **square footage**, number of **bathrooms**, **bedrooms**, **address**, and a **photo** of the house. Tim proceeds to click “contact seller” and is then prompted to sign-in to our website. While registered **buyers** have the ability to add **listings** to a **favorites** list, an unregistered **buyer** must register before he or she can use the **favorites** list.
- 2.) **Registered Buyer:** Sally is a registered **buyer** on our website. She can search and browse our page to look for any house may interest her. For example, not only can she see the basic information, such as the **address** of the house, and the contact information of the user, but she can also view a detailed **description** and features of that house. Registered **buyers** will need to provide more personal information to sellers. They have to register with their **phone number** and **email address** as well. Even more, they need to provide their **name**, so the **buyers** have more reliable contact information, making a deal or arrangement between the **buyer** and **seller** possible. Besides these features, registered **buyers** get a **favorites** list. This means Sally can review the houses she has seen before, and also add new houses into her **favorites** list.
- 3.) **Registered Seller:** Morgan is a registered **seller**. She can search and browse our page to look for any customers that may be interested in a house offered by her. For example, a registered **seller** can see the information that all other users on this website can see, such as the **address** of the house, the contact information of the user, a detailed **description**, and features of that house. Registered **sellers**, like Morgan will need to provide extra personal information upon registering. For example, Morgan will have to provide her **phone number**, **email address**, and a **license number**. The most important feature for the **sellers** is that they are allowed to post **listings**. A **listing** provides the information about the **address** of the house, **price**, general **description**, **pictures** of the house, and more, which unregistered and registered **buyers** can then view.
- 4.) **Admin:** Bob is an **admin** of this website. He can view **listings** and browse, like unregistered **buyers** or **sellers** of our website. However, Bob doesn’t need the features that unregistered **buyers**, registered **buyers**, and registered **sellers** have. The **admin** has the power to manage the website. For instance, Bob can delete contents that are inappropriate to show, and block users that commit illegal or profane acts on our website. The **admin**, like Bob, should have contact information on our website to ensure people

who visit can contact him or her. Therefore, there will be a link with **admin's** contact information on the bottom of website for people to contact.

Data Definition

1. Buyer
 - a. Name
 - b. Phone Number
 - c. Email
 - d. Favorites List
 - e. Is Registered (boolean)
2. Seller
 - a. Name
 - b. Phone Number
 - c. Email
 - d. Profession
 - e. Id (Drivers License or Real Estate License)
3. Listing
 - a. Title
 - b. Address
 - c. Description
 - d. Price
 - e. Photo
 - f. Bedrooms
 - g. Bathrooms
 - h. Square footage
4. Admin
 - a. Name
 - b. Email

Functional Requirements

1. The website shall be scalable.
2. Clicking on a picture shall zoom-in on it.
3. **All Users** shall be able to **search** listings.
4. **All Users** shall be able to **browse** listings.
5. **All Users** shall **sign in** to the website in order to access seller contact information.
6. **All Users** shall be redirected to the home page by clicking the website logo.
7. **Registered Buyers** shall be prompted to register their **name**, **phone number**, and optional **email**.
8. **Registered Buyers** shall be able to add and remove listings to their **favorites**.
9. **Registered Buyers** shall be able to access realtor contact information.
10. **Registered Buyers** shall have a **view listings history**.
11. **Registered Buyers** shall have a **dashboard** to easily message **Registered Sellers**.
12. **Registered Buyers** shall have the option leave contact information public or private.
13. **Registered Sellers** shall have a **view listings history**.
14. **Registered Sellers** shall be prompted to register their **name**, **email**, **phone number**, and **license** number for verification.
15. **Registered Sellers** shall be able to list and edit property information such as **description of house**, **price**, and **photos**.
16. **Registered Sellers** shall be able to declare what type of **profession** they are - realtor/landlord/real estate agent/etc.
17. **Registered Sellers** shall have a **dashboard** to easily message potential buyers.
18. **Registered Sellers** shall have a **dashboard** to manage their own sales.
19. **Administrators** shall be able to remove outdated content.
20. **Administrators** shall be able to edit or remove inappropriate content.
21. **Administrators** shall be able to remove accounts proven to be fraudulent.

Non-Functional Requirements

1. Application shall be developed and deployed using class provided deployment stack
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks must be explicitly approved by Anthony Souza on a case by case basis.
3. Application shall be hosted and deployed on Amazon Web Services as specified in the class
4. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.
5. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed
6. Data shall be stored in the MySQL database on the class server in the team's account
7. Application shall provide real-estate images and optionally video
8. Maps showing real-estate location shall be required
9. Application shall be deployed from the team's account on AWS
10. No more than 50 concurrent users shall be accessing the application at any time
11. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
12. The language used shall be English.
13. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.
14. Google analytics shall be added
15. Messaging between users shall be done only by class approved methods and not via e-mail clients in order to avoid issues of security with e-mail services.
16. Pay functionality (how to pay for goods and services) shall not be implemented.
17. Site security: basic best practices shall be applied (as covered in the class)
18. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
19. The website shall prominently display the following text on all pages "SFSU Software Engineering Project, Fall 2017. For Demonstration Only". (Important so as to not confuse this with a real application).

Competitive Analysis

| Features | Zillow | Trulia | Realtor | Redfin | BayRealEstate |
|----------------|--------|--------|---------|--------|---------------|
| Browse | + | + | + | - | + |
| Search | + | + | + | + | + |
| Save Favorites | + | + | + | + | + |
| Saved Searches | + | + | + | + | - |
| Post Listing | + | + | + | - | + |
| Search History | - | + | - | - | + |

+ Feature Exists; ++ Superior; - Does Not Exist

The real top competitors of our website are Zillow, Trulia, and Realtor, as these sites are listed at the top in most visited real estate websites. They share some important aspects that make it easy for users to search for homes; the features listed in the table above are some features that we want to match when compared with these competitors. Since we plan to include a way for registered sellers to post listings, Redfin is less of a concern, as it is more directed towards utilizing their own real estate agents rather than individual sellers. The table also shows that only Trulia has the feature of a search history, something we plan to add for the convenience of users. It is currently undecided whether we will be adding a Saved Searches function as the other sites have.

High-Level System Architecture

We use AWS(Amazon Web Services) for our server and My SQL for our database. The front-end uses Chrome Developer Tools, Node.js for front-end framework, while the back-end uses Express.

Amazon Web Services: Our web will be hosted on Amazon Cloud, which offers security cloud storage, file backup, file sharing etc.

My SQL: MySQL is an open source relational database management system (RDBMS) based on Structured Query Language. MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. (<http://searchoracle.techtarget.com/definition/MySQL>)

Linux: a Unix-like, open source and community-developed operating system for computers, servers, mainframes.

Chrome Developer Tools: a set of debugging tools built in Google Chrome. DevTools provide deep access into internals of the browsers and web application.

Express: a server side framework. It helps to organize the application's routing and use many templating solution to decrease the amount of work. Addition, Express.js, the “E” means software stack. Every software stack includes an operating system, database, server, and development platform.

Node.js: a platform built on Chrome's JavaScript. **Node.js** uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. (https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm)

BootStrap: a free collection of tools for creating a websites and web applications. It contains HTML and CSS-based design templates.

jQuery: a javascript library. It manipulates HTML to create movement and animation on the webpage. jQuery simplifies the HTML's client-side scripting, thus simplifying Web 2.0 applications development.

Github: a web-based Git or version control repository and Internet hosting service

Github provides a platform to share an entire project as repo among a team or to the whole world.

Our Team

1. Jason Cromer: Team Lead
2. Artsem Holdvekht: Back-end Lead
3. Jiawen Zhu: Front-end Lead
4. Jordan Leong: Front-end
5. Chen-Feng Huang: Front-end
6. David Hoang: Back-end

Checklist

1. Team decided on basic means of communication: DONE
2. Team found a time slot to meet outside of class: DONE
3. Front and Back end team leads chosen: DONE
4. Github Master chosen: DONE
5. Team ready and able to use the chosen Back and Front end frameworks: ON TRACK
6. Skills of each team member defined and known to all: DONE
7. Team lead ensured that all team members read the final m1 and agree/understand it before submission: DONE